Assembling thermal circuits

MATLAB / Octave files:
`t05CubeFBAss.m;`
`fTC2SS.m`
`fTCAssAll.m`
`fSolRadTiltSurf.m`
`fReadWeather.m`

Objectives:

- Assemble complex circuits.
- Check the assembled circuits.

1 Defining the problem of circuit assembling

Given a number of thermal circuits, $TC_1, TC_2, \ldots, TC_n$, and knowing that some of their nodes are common, find the assembled circuit TC (Figure 1).

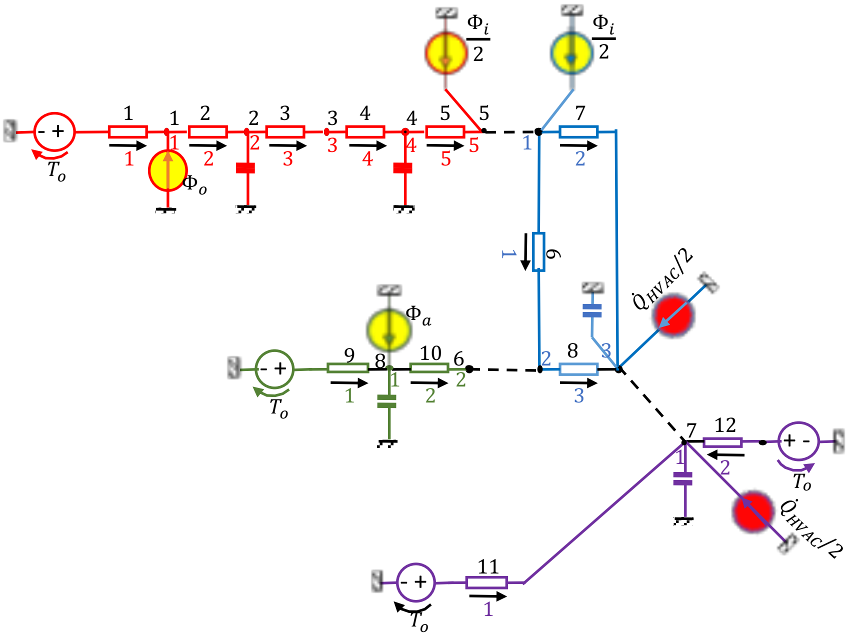


Figure 1 Example of the problem of assembling thermal circuits: given four circuits, assemble them knowing the common nodes. The heat-flow sources and the capacities in the nodes are divided for each circuit

From conservation of energy, it results that if there is a flow source in the node, it needs to be the sum of the sources of each circuit; a simple solution is to be divided equally by each circuit. From conservation of mass, it results that if there is a common capacity in the node, it needs to be the sum of the sources of each circuit; a simple solution is to be divided equally by each circuit (Figure 1).

To exemplify the procedure, we will use a toy model representing a building formed by an insulated concrete wall and a glass wall. The room is ventilated and the temperature is controlled by a P-controller to which additional load is added (Figure 3). The toy model is used to show specific aspects of the assembling procedure, not for correctness of the modelling.

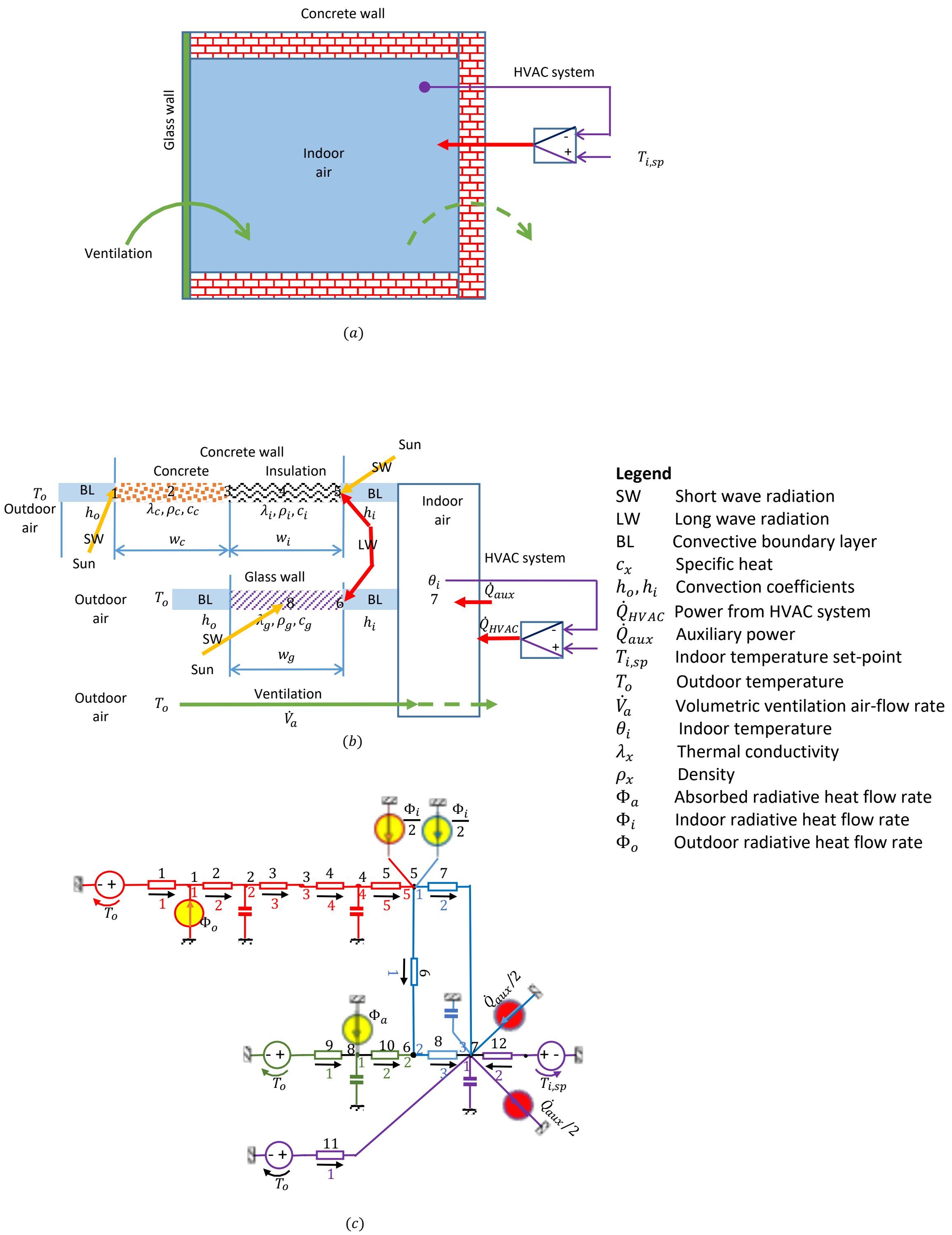
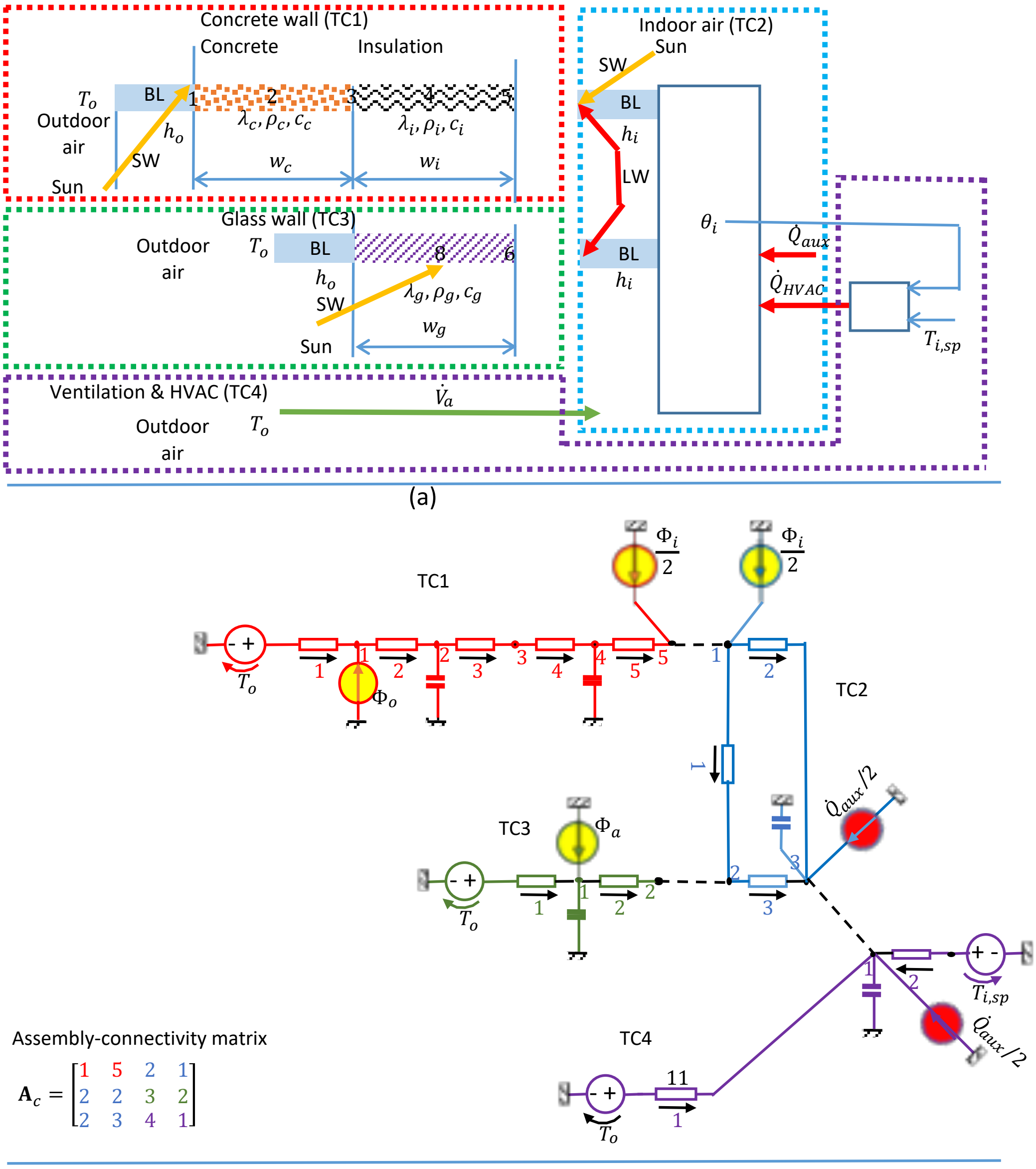


Figure 2 Example of the problem of assembling thermal circuits: a) the physical system – a simple room; b) the conceptual model of the room composed by four elements: 1) concrete wall, 2) indoor air, 3) glass wall, 4) ventilation and HVAC system; c) assembling the four elementary models; the flow sources and the capacities in the nodes are divided in equal parts for each circuit



(b)

$$\mathbf{A_1} = \begin{bmatrix} 1 & & & & \\ -1 & 1 & & & \\ & -1 & 1 & & \\ & & -1 & 1 & \\ & & & -1 & 1 \end{bmatrix} \quad \mathbf{G_1} = \text{diag}\begin{bmatrix} G_{wo} \\ 2G_c \\ 2G_c \\ 2G_i \\ 2G_i \end{bmatrix} \quad \mathbf{b_1} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{C_1} = \text{diag}\,[0 \quad C_c \quad 0 \quad C_i \quad 0]^T$$
$$\mathbf{f_1} = [1 \quad 0 \quad 0 \quad 0 \quad 1]^T$$

$$\mathbf{A_2} = \begin{bmatrix} -1 & 1 & \\ -1 & & 1 \\ & -1 & 1 \end{bmatrix} \quad \mathbf{G_2} = \text{diag}\begin{bmatrix} G_{LW} \\ G_{wi} \\ G_{gi} \end{bmatrix} \quad \mathbf{b_2} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{C_2} = \text{diag}\,[0 \quad 0 \quad C_a/2]^T$$
$$\mathbf{f_2} = [1 \quad 0 \quad 1]^T$$

$$\mathbf{A_3} = \begin{bmatrix} 1 & \\ -1 & 1 \end{bmatrix} \quad \mathbf{G_3} = \text{diag}\begin{bmatrix} G_{g*} \\ 2G_g \end{bmatrix} \quad \mathbf{b_3} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\mathbf{C_3} = \text{diag}\,[C_g \quad 0]^T$$
$$\mathbf{f_3} = [1 \quad 0]^T$$

$$\mathbf{A_4} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \mathbf{G_4} = \text{diag}\begin{bmatrix} G_v \\ G_c \end{bmatrix} \quad \mathbf{b_4} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\mathbf{C_4} = \text{diag}\,[C_a/2]^T$$
$$\mathbf{f_4} = [1]^T$$

(c)

Figure 3 Elementary models: a) schematic representation; b) 4 thermal circuits; c) algebraic representation

We would like to construct separate models for concrete wall, glass wall, ventilation, and room air and to assemble them into one model (Figure 3).

2 Numbering the assembled circuit

2.1 Numbering elementary circuits

In principle, the numbering of the nodes and branches can be done arbitrarily. The connections are indicated by the oriented incidence matrix **A**. Since numbering becomes tedious for large circuits, the following rules may be adopted (Figure 3b):

- number the nodes in order (e.g. from left to right);
- number the branches in increasing order of nodes and orient them from the lower to the higher node. Note: reference temperature is node 0.

2.2 Numbering the assembled circuit

When assembling the thermal circuits, some nodes are put in common. Therefore, the number of nodes in the assembled circuit will be smaller than the sum of the nodes of elementary circuits. The number of branches will not change. The nodes and the branches of the assembled circuit will be in the order of assembling (Figure 1c, Table 1 & Table 2).

Table 1 Local and global indexing of nodes

Thermal circuit	TC1	TC2	TC3	TC4
Local node index	1 2 3 4 5	1 2 3	1 2	1
Global node index	1 2 3 4 5	5 6 7	8 6	7

Table 2 Local and global indexing of branches

Thermal circuit	TC1	TC2	TC3	TC4
Local branch index	1 2 3 4 5	1 2 3	1 2	1 2
Global branch index	1 2 3 4 5	6 7 8	9 10	11 12

The assembling of the circuits is indicated by the assembling matrix. Each row of this matrix has four elements that indicate two nodes that will be put together:

- number of circuit 1
- node of circuit 1
- number of circuit 2
- node of circuit 2

For our example, the assembling matrix is:

$$\mathbf{A}_{ss} = \begin{bmatrix} 1 & 5 & 2 & 1 \\ 2 & 2 & 3 & 2 \\ 2 & 3 & 4 & 1 \end{bmatrix}$$

```
% Assembling matrix
AssX = [1 nt 2 1;...% TC1#5 <- TC2#1
        2 2 3 2;... % TC2#2 <- TC3#2
        2 3 4 1];   % TC2#3 <- TC4#1
```

The description of the assembled circuit, given by the cell array $TC = \{TC_1, \dots, TC_n\}$ of cell arrays $TC_i = \{\mathbf{A}_i, \mathbf{G}_i, \mathbf{b}_\mathrm{i}, \mathbf{C}_i, \mathbf{f}_i, \mathbf{y}_i\}$ and the assembling matrix $\mathbf{A}_{ss}$ (Figure 3c) contain all the necessary information for obtaining the assembled circuit.

3 Implementation

The assembling is implemented in `function [TCa, Idx] = fTCAssAll(TCd, AssX)` in the file `fTCAssAll.m`.

The inputs are

- the cell array `TCd` containing cell arrays of the circuits. For example:
```
TCd{1} = {A1,G1,b1,C1,f1,y1};
TCd{2} = {A2,G2,b2,C2,f2,y2};
TCd{3} = {A3,G3,b3,C3,f3,y3};
TCd{4} = {A4,G4,b4,C4,f4,y4};
```

- the assembling matrix `AssX` with 4 elements on each line: the number of the circuit, the local number of node of the 1st circuit and the number of the circuit, the local number of node of the 2nd circuit.
```
% Assembling matrix
AssX = [1 nt 2 1;...% TC1#5 <- TC2#1
        2 2 3 2;... % TC2#2 <- TC3#2
        2 3 4 1];   % TC2#3 <- TC4#1
```

The outputs are:

- the cell array of the assembled circuit: `TCa = {A,G,b,C,f,y};`
- the indexes of the local circuits and of the global assembled circuit: `Idx`

3.1 Obtaining the global indexes of the assembling matrix

In order to indicate the common nodes of the circuits, it is convenient to give the `AssX` with 4 elements on each line:

1. the number of the 1st circuit,
2. the local number of the node of the 1st circuit,
3. the number of the 2nd circuit
4. the local number of the node of the 2nd circuit.

We need to obtain an assembling matrix Ass of two columns of global disassembled nodes that are put in common. For our example (Figure 5),
6 6.1

- the node 5 (i.e. the final node) of TC1 is put in common with the node 1 of TC2, which has the global value 5+1=6 (5 = number of nodes of TC1, 1 = local index in TC2);
- the node 2 of TC2 (global value 5+2) is put in common with the node 2 of TC3 (global value 5+3+2, where 5 = number of nodes of TC1, 3 = number of the nodes of TC2, 2 local index in TC3);
- the node 3 of TC2 (global value 5+3) is put in common with the node 1 of TC4 (global value 5+3+2+1=11, where 5 = number of nodes of TC1, 3 = number of the nodes of TC2, 2 = number of nodes in TC3, 1 = local index in TC4);

4 Numerical experiments and discussion

4.1 Check the model

Check the local and global indexes of the models.

```
disp('Local and global indexes of nodes');      disp(Idx{1})
disp('Local and global indexes of branches');   disp(Idx{2})
```

4.2 Repeat the experiments from tutorials 3 and 4

Compare the values obtained with `t03CubeFB.m` and `t04CubeFBmesh.m`.